# WP 2.1+2.3 Project Deliverable

# Report on available developer tools, Report on available system capabilities (hardware)

| | |
|---|---|
| Project Number | IST-2000-29266 |
| Project Title | Virtual Real Time Fire Emergency Simulator |
| Deliverable Type | Report |
| Deliverable Class | Internal |

| | |
|---|---|
| Deliverable Number | D2.1 + D2.3 |
| Title of Deliverable | Report on available developer tools [WP2.1]<br>Report on available system capabilities (hardware) [WP2.3] |
| Nature of the Deliverable | Report |
| Contributing WPs | WP2.1, WP 2.3 |
| Contractual Date of Delivery | 30. November 2001 |
| Actual Date of Delivery | 11. December 2001 |
| URL | www.virtualfires.org |
| Authors | Sascha Schneider (FIGD), Michael Schmidt (FIGD), Thomas Reichl (SiTu), Kai-Mikael Jää-Aro (KTH), Björn Sjögreen (KTH), Gert Svensson (KTH), José Luis Ajuria (EUVE), Christian Redl (CD), Wilhelm Brandstätter (CD) |
| Contact Details | Institute for Structural Analysis / SiTu Research<br>Univ. Prof. Dipl.-Ing. Dr. techn. Gernot Beer<br>Lessingstrasse 25/II<br>8010 Graz / Austria<br>Tel.: +43 316 8736180<br>Fax: +43 316 8736185<br>Email: gernot.beer@ifb.tu-graz.ac.at |

| | |
|---|---|
| Abstract | The summarized results of WP2.1 and WP2.3 are presented in this report. The available developer tools and the hardware of all project partners are listed to give an overview. |
| Keywords | Developer Tools, Software, Hardware, Simulation, Visualization |

# Contents

# 1 Available developer tools

## 1.1 CD

### 1.1.1 Numerical simulation part

There are several Computational Fluid Dynamics (CFD) software packages available at CD. They all share common features with the exception of the last programme (ICE) described below.

The “classical” flow solvers (FIRE, STAR-CD, FLUENT) are based on the Navier-Stokes equations with turbulence effects included via the k-e model. Pressure-velocity coupling is achieved by variants of the SIMPLE-algorithm and the finite-volume method is used for special discretisation on irregular computational meshes. Time integration is done in an implicit fashion.

On the other hand ICE solves a discretised form of the discrete Boltzmann equation utilizing an explicit time integration scheme and regular Cartesian grids (lattices).

#### *1.1.1.1 FIRE*

FIRE is a CFD program developed by AVL List GmbH, Graz, Austria, [CD1] and was specifically designed to simulate flow and combustion processes in internal combustion engines. Due to its specialisation on internal combustion, FIRE is not well suited for general purpose applications. Furthermore the future development of FIRE seems to be uncertain.

#### *1.1.1.2 STAR-CD*

Another commercially available CFD program specifically designed for automotively applications is STAR-CD from the Adapco group [CD2].

#### *1.1.1.3 FLUENT*

The general purpose CFD code FLUENT [CD3] has been already used successfully at CD for the simulation of tunnel fire hazards.

It offers the possibility to implement user defined functions, which enable the integration of the proposed VIRTUALFIRES combustion model. This code will be used for all a-priori simulations, as discussed below.

Another major advantage of FLUENT when compared to e. g. FIRE is the possibility of using unstructured meshes and automatic mesh generators. Furthermore the good and fast support from the vendor has to be emphasized. No wonder, that in the meantime FLUENT has become the most widely used CFD program.

An intensive validation and test phase will include comparisons between FLUENT results and ICE predictions.

#### *1.1.1.4 ICE*

I(nnovative) C(omputational) E(nvironment) [CD4] is a fluid dynamics simulation program based on the Lattice-Boltzmann method.

The software is under continuous development at CD. The theoretical background for a real time combustion model has been already defined and is ready to be implemented.

The major advantages of ICE are as follows:

- The time explicit algorithm avoids the solution of large systems of linear equations.
- Most of the floating point operations can be done on a local basis and hence efficient parallelisation is possible.
- As ICE works on strictly regular, orthogonal grids, simple data decomposition is sufficient for parallelisation.

### 1.1.2 General development part

The selection of the development tools for the simulation software requires special attention. Four main points (programming language, compilers, parallelization concepts and parallel communication software) are of utmost importance.

#### *1.1.2.1 Programming language*

There are two alternatives for developing CPU intensive high-performance simulation software: C++ and FORTRAN.

The motivation for using FORTRAN rather than C++ is as follows.

FORTRAN90/95 contains almost every feature considered as important for object oriented programming (e.g. abstract data types, encapsulation, inheritance, generic programming, etc.).

FORTRAN90/95 includes a very powerful and failsafe array syntax.

FORTRAN90/95 is still seen to be the most widely used language for technical and scientific computing and therefore it is easy to share and communicate with other scientists and developers.

Long term experience for developing and parallelising FORTRAN based simulation software is existing at CD.

Extremely good performance data of programs written in FORTRAN have been measured on COMPAQ DEC Alphaserver systems as used within CD. Similar efficiency is expected on the IBM SP system installed at the Centre for Parallel Computers, KTH Stockholm [CD5].

All simulation software developed within VIRTUALFIRES hence will be written in strict adherence to the FORTRAN90/95 standard [CD6] without using additional features offered by various compilers. No scientific or mathematical libraries will be used. This should guarantee hardware platform independence.

#### *1.1.2.2 Compilers*

For compilation of the user-defined functions of FLUENT the proprietary internal FLUENT C-compiler will be used. The native Compaq FORTRAN90/95 compiler [CD7] for Tru64 UNIX Alpha systems will be used for the further development of ICE and hence the integration of the combustion module.

For simulations on the IBM SP of KTH the native IBM XLF90 compiler [CD8] is proposed. As mentioned before, ICE is written standard Fortran90/95 and hence porting from the COMPAQ DEC Alpha server to the IBM SP should represent no problem at all.

#### *1.1.2.3 Parallelization concepts*

High Performance FORTRAN

The High Performance FORTRAN (HPF [CD9], standardised 1993) is a data parallel model for MIMD (SMP) systems. It is based on Fortran and directives, which are used for the definition of data distribution and processor topologies and to give the compiler support for the detection of loop concurrencies and global operations.

#### *1.1.2.4 Message Passing Interface*

An alternative to HPF is the message passing programming model, where the communication between the processors is performed by sending and receiving messages. The Message Passing Interface (MPI [CD10], standardised 1995) is a parallel programming library that specifies subroutine calls which support communication operations. Language bindings are available to Fortran and C/C++. The data distribution and communication must be defined explicitly by the programmer.

#### *1.1.2.5 Selection of Parallelization Model*

Since the hardware platforms used in the VIRTUALFIRES project are different from the architectural point of view, special care has to be taken to the choice of the parallel programming model. The COMPAQ DEC Alphaserver installed at CD is a symmetric multi processing systems with one single memory shared by all processors. On the contrary the IBM SP at KTH represents a distributed memory system, where each processor accesses its own memory. Although program development based on HPF is much easier when compared to MPI, experience has shown, that MPI out performs HPF significantly when a large number of processors is to be used. Taking into consideration the expected extreme demands on CPU resources by the real time VIRTUALFIRES simulations, hence MPI is the preferred choice.

#### *1.1.2.6 MPI Implementation*

Although standardized, there are several implementations of MPI available. MPICH [CD11] from Argonne National Laboratory and Mississippi State University is a reference implementation of the MPI standard. It usually has the most complete and up-to-date information. The current version is 1.2.2, which was released in August 2001. COMPAQ MPI [CD12] is an optimised version of MPICH functionally compatible with MPICH version 1.1.1., and this version will be used for all parallel software development work at CD.

At KTH an IBM MPI implementation [CD13] as well as the MPICH version 1.2.0 is available. Hence installation of the parallel VIRTUALFIRES software should be straightforward.

## *1.2 KTH*

### 1.2.1 Numerical simulation part

6 licenses Fluent 5.2, available for both SP and SGI.

Fluent requires one licence per started process, which means that the parallelism is currently limited to 6 processors. This is sufficient for the development stage. When the code is ready for a realistic application, the possibility of acquiring more licences, or perhaps a parallel license should be investigated.

On SP:

- XL Fortran for AIX version 7.1
- IBM C++ version 3.6.6

On SGI:

- MIPS C/C++ compiler
- MIPS Fortran compiler

On Sun:

- f77, FORTRAN 3.0.1 from Sun
- gcc

Parallel programming:

- MPI

Previous practical experiences with porting Fortran code have shown that the XLF compiler on the SP is in general more restrictive about the Fortran extensions and syntax than the MIPS compiler on SGI. On the other hand XLF often makes a better job at correcting minor mistakes, such as, e.g., uninitialized variables, which potentially come out as errors when run on SGI. The code should be continuously tested out on all systems on which it should run, during the development process.

### 1.2.2 General development part

Prototyping can preferably be done in COVISE, with the COVER plugin for display [KTH1]. COVISE runs on both Irix and Linux, reads Fluent-format data files and has support for collaboration with multiple distributed users.

If performance increases turn out to be necessary, software can be reimplemented in OpenGL, preferably using OpenGL Volumizer and OpenGL Performer to simplify development [KTH2. OpenGL is available for Irix and Linux, as well as a number of other operating systems. OpenGL Volumizer is only available for SGIs with hardware texture support. OpenGL Performer is available for both Irix and Linux.

In this latter case we recommend using VR Juggler [KTH3] for support of VR systems. VR Juggler is free software and is available for Irix, Linux and other operating systems. VR Juggler supports output to both CAVE-type systems as well as HMDs.

For distribution in VR we recommend the use of CAVERNsoft G2 [KTH4]. CAVERNsoft is free software and available for Irix, Linux and a number of other operating systems. In particular it has support for the rapid transfer of very large datasets. CAVERN modules have support for the construction of 3D GUIs.

OpenGL Performer is able to read geometry files in a number of formats including DXF, STL, FLT, VRML and 3DS.

Geometry can also be converted, including polygon optimisation, with PolyTrans [KTH5].

RasDaMan is a database for multidimensional raster data including some tools for compression. It might be useful for storing CFD data on regular grids. Irregular grids of some forms may also be possible [KTH6].

A more complete overview of resources at PDC is available at [KTH14].

## *1.3 SiTu*

### 1.3.1 Numerical simulation part

#### *1.3.1.1 TunnelVis*

For input of geometrical data and generation of meshes there is one software package available (TunnelVis), that is described below.

TunnelVis is a prototype of a pre-processor for the generation of tunnel-geometry data and generation of the meshes for the CFD- simulation, developed by Bernhard Mathis and Thomas Reichl. Its sourcecode is fully available and can be adapted for special features for VIRTUALFIRES.

### 1.3.2 General development part

Currently there are the following developer tools available at SiTu:

Microsoft Visual Studio 6.0 is used for the development of Windows software packages. Digital Fortran Visual Studio 6.5 is also available but not used by Gunther Lenz and Thomas Reichl.

Gernot Opriessnig used RapidApp on Irix platform. It is an GUI developer tool which is used for programming the software package TVS described below.

Considering platform independence, we would suggest the following tools:

#### *1.3.2.1 Compiler-system*

MS Visual Studio with the Intel/Kai C++ compiler plug-in on Windows platform and xemacs or K-Develop with the Intel/Kai C++ compiler on Unix/Linux platform.

The Intel/Kai compiler is close to the ISO C++ standard and comes with the same libraries on all supported platforms.

For our part in WP2 and WP3 there seems currently no necessity for using Fortran.

#### *1.3.2.2 GUI-library*

Trolltech Qt 3.0 Windows as described below by FIGD should be used for being platform independent.

#### 1.3.2.3 Standard graphics library

OpenGL 1.2/1.3 is the standard for 3D graphic visualisation. More details are described below by FIGD.

#### 1.3.2.4 Parallel processing

We agree with the choice of CD of using MPI for parallel processing software.

#### 1.3.2.5 TVS visualisation

The TVS [SiTu1] was developed by Gernot Opriessnig and runs on an Octane MXI. Its purpose is to render data from the BEFE [SiTu2] (finite element/ boundary element software developed by Gernot Beer) in real time. It supports the visualization of scalar surface data via contour plots, vectorial data is displayed as arrows. Scalar volume data is shown via simulated fog, whose density is proportional to the data value. Animation of load steps is also supported. Navigation is done mainly via SpaceBall, although there is rudimentary support for using a standard mouse. Stereo vision is currently supported by using CrystalEyes shutter-glasses on the Octane, support for i-glasses is under development.

The biggest visualized meshes consisted of about 10000 elements and incorporated 50 load steps. Import-filters for BEFE and formatted ASCII-lists are implemented. Source code is fully available. Required hardware platform for TVS is an sgi Octane MXI with IRIX 6.5.

## 1.4 FIGD

### 1.4.1 General development part

There are several tools available on the market for developing the visualization part of the VIRTUALFIRES project. Some of them are already available at FIGD.

To select among all the desired tools several criteria were considered. In the first the chosen tools should guarantee a good performance and scalability. To be as independent as possible from the underlying operation system they should allow cross platform portability and easy handling.

Furthermore the distribution of the selected tools and of course the available support, know how and documentation play an important role for the decision which tools are taken into account.

In the following section an overview of the available developer tools at FIGD are presented:

#### 1.4.1.1 Compiler system / programming language

##### 1.4.1.1.1 Microsoft Visual C++ 6.0 [FIGD1]

This is probably the most used C++ compiler for MS Windows. In fact most of the commercial applications for these operating systems are developed using this compiler. It is a complete development tool with an IDE that integrates debugging and visual development tools. It also includes specific components to develop databases, internet applications and of course MFC and ATL based applications.

MS VC allows simplified programming with COM+, Microsoft SQL Server™ 7.0, SQL Server-compatible Microsoft Data Engine (MSDE) and has support for Microsoft Transaction Server (MTS) and the Internet Information Server (IIS), Active Template Library (ATL). Furthermore it supports the ANSI/ISO C++ specification, industry-standard MFC, Visual SourceSafe® 6.0, IntelliSense® Technology, integrated Visual Database Tools (for Microsoft SQL Server 6.5+ and Oracle 7.3.3+ databases control) and has many new and enhanced wizards (e.g. OLE DB wizards) together with remote SQL debugging and the Visual Modeler.

#### 1.4.1.1.2 Borland C++ Builder 5.0 [FIGD7]

Borland C++ is a productive environment to develop Windows applications in a fast and easy way. It is known to be one of the best ANSI-C++ compliant compilers currently available in the market. Supports the building of Windows applications using the Win32 API, COM components, ActiveX and OLE.

It allows advanced Smart Debugging, has a powerful ANSI/ISO C++ compiler, a simplified CORBA Distributed Object Development, ANSI/ISO Standard C++ Library (STL), ActiveInsight, Microsoft Visual C++ compatibility, TeamSource and Borland Translation Suite. Further Features are Visual Component Library, WebBroker, InternetExpress, Borland MIDAS, InterBase Express (IBX), ADOExpress, InterBase-embedded RDBMS and BusinessInsight.

#### 1.4.1.1.3 Intel C++ Compiler 5.0 [FIGD8]

The Intel® C++ compiler 5.0 is an important part of making software run at top speeds on Intel® 32-bit and 64-bit computers. Through the use of new compiler optimizations, such as the prefetch instruction and enhanced support for the Intel® Pentium® 4 and Itanium™ processors, the Intel C++ compiler 5.0 can deliver dramatic application performance improvements. Using features like the unique Whole-Program Optimization and Profile-Guided Optimization can lead to even more dramatic application performance. The advantage of these optimization techniques comes from the combination of innovative Intel hardware and software technology that delivers performance and value for systems based on the Intel® processors, including the Pentium 4 and Itanium processors.

It has Microsoft Visual C++ Compatibility and plugs into the IA-32 Microsoft Visual Studio development environment. Additionally it has good floating-point instruction throughput, Data prefetching, Interprocedural Optimization (IPO), Profile Guided Optimization (PGO) and an enhanced Debugger (EDB) from Intel.

#### 1.4.1.1.4 GCC 3.0 [FIGD9]

GCC is used to create everything from Linux and its various BSD Unix cousins to higher-level software such as the Apache Web server, the Gnome user interface and the Jabber instant-messaging software. And it can run on and create software for more than 40 different chip families.

Some new features, which are available through GCC are general code optimizer improvements, new languages and language specific improvements, and finally new targets and target specific improvements. Last but not least GCC is available on almost all operation systems.

### *1.4.1.2 GUI-library*

#### 1.4.1.2.1 Trolltech Qt 3.0 Windows [FIGD2]

Qt is a cross-platform C++ GUI application framework. It provides application developers with all the functionality needed to build state-of-the-art graphical user interfaces. Qt is fully object-oriented, easily extensible, and allows true component programming. Since its commercial introduction in early 1996, Qt has formed the basis of many thousands of successful applications worldwide. Qt is also the basis of the popular KDE Linux desktop environment, a standard component of all major Linux distributions.

Some of the features QT supports are object orientation (widget encapsulation), component support (signals/slots concept), detailed HTML online documentation, easy customization, portability (multi-platform toolkit) and internationalization (message translation tables). It is a rich API (around 250 classes) with a full Widget set allowing high performance implementations (fast execution and conservative memory usage) with GUI Emulation (enhanced functionality across all platforms). Furthermore QT can provide a customizable look and feel (Motif, Windows, ...) together with advanced drawing operations (support for any paint device) using 2D/3D graphics rendering (OpenGL Widget).

### *1.4.1.3 3D Graphics API*

#### 1.4.1.3.1 SGI OpenGL 1.2/1.3 [FIGD3]

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

Some keywords regarding supported by OpenGL as graphics API are accumulation buffer, alpha blending, anti aliasing, color index mode, display lists, double buffering, feedback, gouraud shading, immediate mode, materials lighting and shading, pixel operations, polynomial evaluators (NURBS), primitives, RGBA mode, selection and picking, stencil planes, texture mapping, transformation, z-buffering, 3D texturing, BGRA pixel and packed pixel formats, automatic rescaling of vertex normals, specular highlights after texturing, texture coordinate edge clamping, level of detail control and last but not least vertex array enhancements.

# 2 Available system capabilities (hardware)

## *2.1 CD*

### 2.1.1 For simulation

The VIRTUALFIRES software will be developed on COMPAQ DEC Alpha platforms, which are already available at CD.

For the base software development a COMPAQ DEC Alphaserver DS10 (1 processor, 466 MHz, TPP 932 MFLOPS, up to 1 GByte memory) is used.

The parallelisation is done on a COMPAQ DEC Alphaserver ES40 (3 processors, 833 MHz, TPP 1666 MFLOPS/processor, total TPP 5000 MFLOPS, 8 GByte memory).

The hardware at KTH consists of 170 IBM SP processors reaching a TPP of 201 GFLOPS. Text

## *2.2 KTH*

### 2.2.1 For visualization

1 six-wall TAN VR-CUBE [KTH7] with 4-tracker Polhemus FASTRAK Long Ranger [KTH8], eight-channel audio system using

1 SGI Onyx 2, running Irix 6.5, with the following hardware:

- 12 195 MHZ IP27 Processors
- CPU: MIPS R10000 Processor Chip Revision: 2.6
- FPU: MIPS R10010 Floating Point Chip Revision: 2.6
- Main memory size: 4096 Mbytes
- Graphics board: InfiniteReality2 (* 3)
- Iris Audio Processor: version RAD revision 7.0 (* 2)
- The full properties of the hardware of KTH are listed at [KTH9].

1 ImmersaDesk [KTH10] with 2-tracker

Ascension SpacePad [KTH11] using

1 SGI Octane, running Irix 6.5, with the following hardware:

- 2 250 MHZ IP30 Processors
- CPU: MIPS R10000 Processor Chip Revision: 3.4
- FPU: MIPS R10010 Floating Point Chip Revision: 0.0
- Main memory size: 1024 Mbytes
- Graphics board: EMXI
- Iris Audio Processor: version RAD revision 12.0
- Further information is available at [KTH12].

1 Virtual i/o i-glasses HMD [KTH13].

1 RPI/ATG HMS-EYE2 CUSTOM HMD.

### 2.2.2 Potentially useful for visualization/computation

1 PC-cluster with 16 PIII 866 MHz processors, each with 256 MB of memory. Peak performance per CPU is > 800 MFLOPS. No current graphics hardware.

### 2.2.3 For simulation / computation

1 300-processor IBM SP cluster. Total main memory is 115 GByte and theoretical peak performance 204 GFlops.

### 2.2.4 For Data storage

- 1 IBM 3494 Tape robot with 4 IBM 3590 Tape stations
- Total storage capacity of around 30 TB. Software: IBM Tivoli Storage
- Manager and SGI DMF (Data Migration Facility) HSM system.

## *2.3 SiTU*

At the moment there has no new hardware been tested.

TVS is currently being ported to Windows and seems to run on a Dell Inspiron 8100 with a nVidia Geforce2go in stereo-mode. This configuration requires a video-splitter for correct working of the I-glasses HMD.

Other available hardware: sgi Octane-MXI, sgi Visual PC

## *2.4 FIGD*

### 2.4.1 For visualization computation and output

The CAVE technology combines high resolution, high image quality, large screen projection and therefore large field-of-view with a fully immersive output technology. In addition, a CAVE provides the perception of presence, since the user observes the VR scenario in relation to his real body, arms and hands. Navigating through virtual scenes or interacting with virtual objects, a CAVE supports a space impression of 1:1 scale. The 5-sided CAVE installation at IGD [FIGD4] consists of 5 perpendicular stereo projection screens (3 walls, floor and ceiling with 2.4m x 2.4m each). The whole construction is mainly made of wood, in order to avoid any electromagnetic distortion for the tracking system. For the realization of the rear-projection of the floor, we had to lift up the CAVE about 2 meters. For the floor, we installed a 30 mm "paraglass" plate, a special acrylic glass, which is strong enough to carry 10 persons without significant distortion of the light beam and with no observable light absorption. The screens consist of special plastic material and all screens are melted along the corners without any special construction, in order to guarantee shadow free projection. For the projection itself, 5 modified Electrohome Marquee 7500 are used with an image resolution of 1024x1024 pixels with 120 Hz. The images are generated by a 3-pipe SGI ONYX Infinite Reality with 6 CPUs (R10000). In addition, a 3D sound equipment with 4 speakers and several 3D position devices are connected.

The FIGD CAVE can be driven by two different graphic servers with the following features:

On the one hand a SGI Onyx [FIGD5] with 6 Processors at 194 MHz IP25 is available. It consists of

- CPU: MIPS R10000 Processor (Rev 2.6 / 2.5)
- FPU: MIPS R10010 Co-Processor (Rev 2.6 / 2.5)
- Main memory: 1024 Mbytes
- Instruction cache: 32 Kbytes
- Data cache: 32 Kbytes
- Secondary cache: 1 Mbyte
- - 3 Graphics boards (Infinite Reality)

On the other hand a SGI Onyx 2 [FIGD6] with 4 Processors at 195 MHz IP27 is available. It consists of

- CPU: MIPS R10000 Processor (Rev 2.6)
- FPU: MIPS R10010 Co-Processor (Rev 2.6)
- Main memory: 2048 Mbytes
- Instruction cache: 32 Kbytes
- Data cache: 32 Kbytes
- Secondary cache: 4 Mbytes
- 2 Graphics boards (Infinite Reality 2)

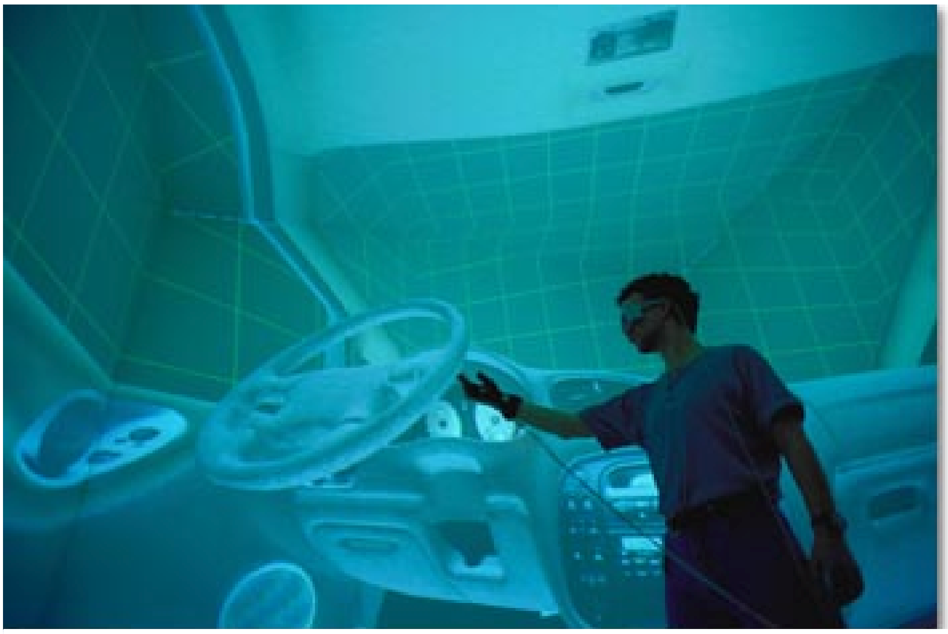

figure 2-0: The CAVE at FIGD – internal view

figure 2-0: The CAVE at FIGD – external view

# Literature/Links


[CD1] http://www.avl.com

[CD2] http://www.cd.co.uk

[CD3] http://www.fluent.com

[CD4] ICE User manual, Christian-Doppler-Laboratory for Applied Computational, Thermofluiddynamics, Leoben, 2001

[CD5] http://www.pdc.kth.se

[CD6] http://www.j3-fortran.org

[CD7] http://www.compaq.com/fortran

[CD8] http://www-3.ibm.com/software/ad/fortran/

[CD9] http://www.crpc.rice.edu/HPFF/

[CD10] http://www.mpi-forum.org

[CD11] http://www-unix.mcs.anl.gov/mpi/mpich/

[CD12] http://www.compaq.com/hpc/software/dmpi.html

[CD13] http://www.rs6000.ibm.com/resource/aix_resource/sp_books/pe/

[FIGD1] http://msdn.microsoft.com/visualc/default.asp

[FIGD2] http://www.trolltech.com/products/qt/windows.html

[FIGD3] http://www.sgi.com/software/opengl/

[FIGD4] http://www.igd.fhg.de/igd-a4/index.html

[FIGD5] http://www.futuretech.vuurwerk.nl/onyx/

[FIGD6] http://www.futuretech.vuurwerk.nl/onyx2/

[FIGD7] http://www.borland.com/bcppbuilder/

[FIGD8] http://developer.intel.com/software/products/compilers/c50/

[FIGD9] http://gcc.gnu.org/

[KTH1] http://www.vircinity.de/d+e/data0801/englisch/e_produkte/xe_fs_prod_cov.html

[KTH2] http://www.sgi.com/software/opengl/,
http://www.sgi.com/software/volumizer/,
http://www.sgi.com/software/performer/

[KTH3] http://www.vrjuggler.org/

[KTH4] http://www.openchannelsoftware.org/projects/CAVERNsoft_G2/

[KTH5] http://www.okino.com/conv/conv.htm

[KTH6] http://www.active-knowledge.de/index.en.html

[KTH7] http://www.tan.de/english/produkte/vrcube.html

[KTH8] http://www.polhemus.com/ftrakds.htm

[KTH9] http://www.sgi.com/onyx2/tech_specs.html

[KTH10] http://www.fakespacesystems.com/pdfs/FS_IDeskC1.pdf

[KTH11] http://www.ascension-tech.com/products/spacepad/

[KTH12] http://www.sgi.com/products/remanufactured/octane/overview.htm

[KTH13] http://www.i-glasses.com/Store/iglasses_lc.php3

[KTH14] http://www.pdc.kth.se/compresc/computers.html